A latent semantic analysis method for ranking the results of human disease search engine

Loi Chan Quan Lam[1], Tran Kim Toai[2], Snasel Vaclav[1]
[1]Department of Computer Science, Faculty of Electrical Engineering and Computer Science, Technical University of Ostrava, Ostrava, Czech Republic
[2]Department of Computer Science, Faculty of Economics, Ho Chi Minh City University of Technology and Education, Ho Chi Minh, Vietnam

Article Info

Article history:

Received Aug 22, 2022
Revised Nov 2, 2022
Accepted Nov 25, 2022

Keywords:

Disease ontology
Latent semantic analysis
Ranking algorithm
Search engine

ABSTRACT

The human disease search engine based on the search query about disease factors (symptom, cause, position happening symptoms, i.e.,) helps users to conveniently diagnose the disease they may have anytime, anywhere. Therefore, the disease results returned by the search engine need to be accurate and ranked reasonably so that users can know which disease has the highest probability for their search query. We propose a method to arrange the returned diseases based on the latent semantic analysis (LSA) technique. This method helps to rank the disease results reasonably and meaningfully because it not only exploits the matching term frequency-inverse document frequency (TF-IDF) scores between the disease factors in the query and the disease results, but it also exploits the implicit relationship between the disease factors in the search query and the disease factors in the disease results.

Corresponding Author:

Tran Kim Toai
Department of Computer Science, Faculty of Economics
Ho Chi Minh City University of Technology and Education
No 1 Vo Van Ngan Street, Linh Chieu Ward, Thu Duc City, Ho Chi Minh, Vietnam
Email: toaitk@hcmute.edu.vn

1. INTRODUCTION

The development of the internet and web ontology has made medical and disease data more and more huge [1], [2]. Many disease search engines appear to make it easier for people to access these data sources [1], [3], [4]. Especially the human disease search engine or medical search engine based on disease factors or symptoms helps people conveniently self-diagnose their diseases [1], [5]. Therefore, the disease results returned by the search engine not only need to be accurate but also must be ranked reasonably so that the users can know the disease having the highest probability they are likely to have [6]. In information retrieval, a common method to rank results is the term frequency-inverse document frequency (TF-IDF) method [7], [8]. This method calculates the importance of words in the query to the result document in order to rank the results [9]. However, this method does not address the relationship between the words in the query and the words in the result document. Another method of ranking disease results is using the bayesian algorithm [10]. This method is based on the superclass of the disease results and the number of diseases belonging to the superclass to calculate the probability of the disease results. The limitations of this method are that if the number of diseases of the superclass has only one disease or very few diseases, it will give a very low probability for the disease results, which will not be correct in the case of the disease results containing several disease factors having a high fit to the disease factors in the query. In this paper, we propose a method to rank disease results of search engines using the latent semantic analysis (LSA)

technique. This method exploits the relationship between disease factors in the query and in the disease results to help the result ranking of the human disease search engine more accurately and avoid the limitations of the result ranking by using the bayesian method or common method.

– Disease ontology

Data from the human disease search engine were extracted from disease ontology (DO). DO is an internet resource for disease knowledge [11]. It was created in 2003 by using the ninth revision of international classification of diseases (ICD-9). The DO was then reorganized based on unified medical language system (UMLS) disease concepts [12]. Currently, the DO terms are continuously being improved and extended. DO has a single structure for disease classification and provides a clear definition for each. A disease has a label, definition, subclass, superclass, and property. The disease property or disease factor includes symptom, cause and location (positions happening symptoms). Figure 1 shows the hierarchy of DO.

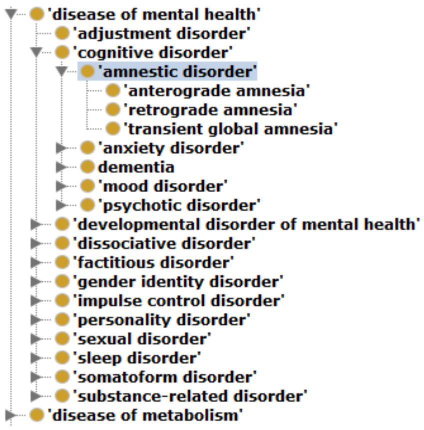

Figure 1. DO hierarchy

– Latent semantic analysis

The proposed method of this paper uses LSA technique. LSA or latent semantic indexing (LSI) is a statistical method that was created in the late 1980s at bell core/bell laboratory by Launder and his team. They defined "LSA as theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text" [13]. Scientific research has proven that LSA is similar to the way the human brain receives meaning from text, and LSA is capable of inferring deeper relationships in text data [14], [15]. LSA starts with creating a term-document matrix; the columns stand for words or terms, and the rows stand for documents. Each entry in the term-document matrix is the TF-IDF score of the word in the document. Next, the singular value decomposition (SVD) technique is applied to the term-document matrix; this step is a key feature of LSA. The matrixes created by this step have a dimensional reduction, and we can exploit the hidden meaning in the text of the document from these matrixes [16].

2. METHOD

The LSA method applies to the human disease search engine as described in Figure 2. In this section, each part of Figure 2 is described in detail. The human disease search engine uses the MySQL database. Data from the disease database were extracted from DO. The search engine accesses MySQL database faster and more flexibly than DO. The disease database includes many tables that store information about all diseases, disease superclasses and subclasses, and all disease factors. The search engine processes so much on the disease definition table, which contains all diseases and their definitions. The definition of disease includes information about the disease and its factors. Disease factors can be a symptom, cause or location (positions happening symptoms). Figure 3 shows a disease definition.

The data of disease definition table consists of column "diseaseID", column "disease label" and column "definition". The column "definition" is extracted into a data frame for tokenized processing. Each row of the data frame is considered as a document. The tokenized processing removes stopwords, punctuation, and lowercase words. Each document is processed into an array of words. TF-IDF algorithm is applied to these arrays.

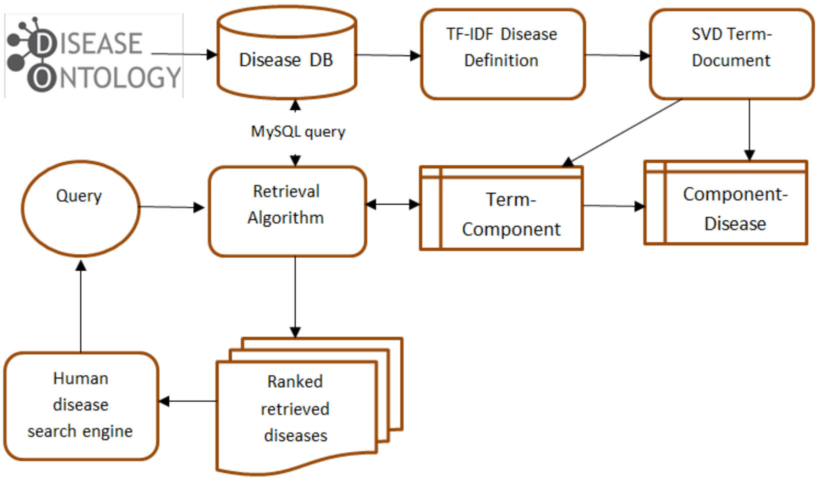

Figure 2. Human disease search engine architecture

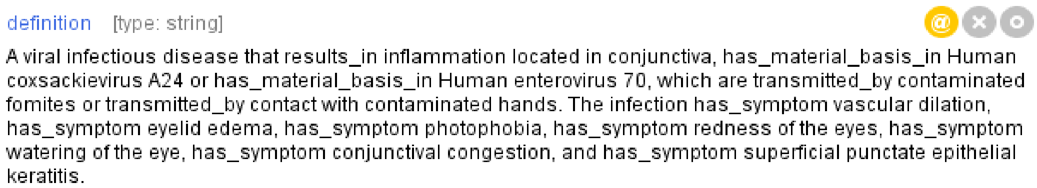

Figure 3. Disease definition

2.1. Query

The human disease search engine supports hint suggestions when users query in the search box of the engine Figure 4. This increases the interaction between the user and the search engine [17]. When users enter the keyword in the search box, the search engine suggests similar keywords in the engine, and users can choose the keyword that suits their intent. These keywords are the disease factors in the engine database, so the returned results will be more precise. Suggested hint also helps users in case they only remember a part of the keyword, they can fill this part in the search box, and the search engine will suggest the full keyword and many other keywords similar to that keyword [17]. Hints are necessary because normal users who do not have much medical knowledge [18], [19] may not be able to enter keywords correctly with the medical expertise contained in the engine database, leading to inaccurate results. For medical experts, hint suggestions will be useful in case they only remember part of the keyword [18], the search engine will fully suggest helping them remember the keyword they need, and they can refer to other similar keywords in the engine.

Figure 5 shows the hint suggestion process of the search engine. The hint suggestion process starts with the user entering keywords into the search engine. After each "space key press event", the search engine will get similar keywords in the cache for suggestions. In case there is no keyword in the cache yet, the search engine will query in the engine database for similar keywords to return to the user and store them in the cache for next time use.

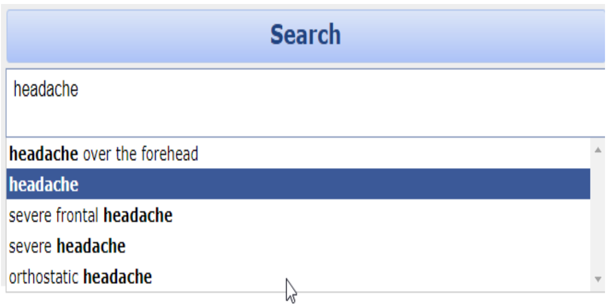

Figure 4. Hint suggestion

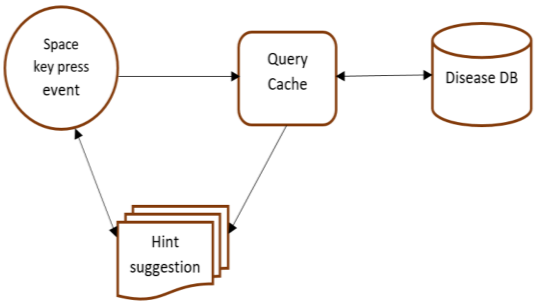

Figure 5. Hint suggestion process

2.2. TF-IDF disease definition

TF-IDF is a technique used in information retrieval to measure the importance of a word to a document in a collection of documents [20]. The TF-IDF of a word is calculated in (1).

$$TF - IDF\ (t, d)\ =\ T\,F\ (t, d)\ \times\ IDF\ (t) \tag{1}$$

where TF is the number of occurrences of that word in the document divided by the number of all words in the document. IDF is calculated in (2).

$$IDF\ (t)\ =\ log\,\frac{|D|}{|\{d: t \in d\}|} \tag{2}$$

|D| is the number of all documents, d is a document, and t is a word or term [21]. In this paper, each disease definition is a document and all diseases in the database are the document collection. TF-IDF technique generates matrix term-document. In this matrix, each word has a score. The term-document matrix of disease data has a large number of columns, about 5,000 columns; processing on this matrix will be computationally expensive. SVD algorithm is applied to this term-document matrix to reduce the dimension and exploit the relationship between words in the document.

2.3. SVD term-document

SVD is a matrix factorization technique to split a matrix into two or three matrices. It is commonly used for dimensionality reduction to make data easier to visualize and extract desired information [22]. Dimension reduction is a process that reduces the number of features [23], so it improves computational efficiency. In addition, dimension reduction also helps to reduce noise and sparsity of the raw features [24]. SVD algorithm is calculated in (3).

$$A = USV^T \tag{3}$$

Where U is a m×r orthogonal left singular matrix, V^T is a r×n orthogonal right singular matrix, S is a r×r diagonal matrix and A is the original matrix [25].

SVD algorithm reduces matrix A from m×m to m×r and r×m (r≤m). In this paper, the original matrix is the term-document matrix. The number of components (r) of the SVD algorithm applied on the term-document matrix of disease data will be the total number of disease classes (super class and subclass) in the disease database (number of disease classes<number of diseases<number of disease words). The results of the SVD algorithm applied to the term-document matrix of disease data create a word-component matrix (U matrix, Figure 6) and a component-disease matrix (V^T matrix, Figure 7).

	activity	acuity	acute	acvr1	adaptor	additional	adducted	ademona	adenofibroma	adenoid
component1	0.000687	0.000160	0.009563	0.000270	0.000042	0.000081	0.000068	0.000014	0.000054	0.002852
component2	0.003525	0.000914	0.017202	0.002197	0.000698	0.000307	0.000480	0.000423	0.001002	-0.000314
component3	0.006424	0.001305	0.037185	-0.000329	-0.000285	0.000846	0.000803	-0.000147	-0.000114	0.002703
component4	0.002037	0.001106	0.001214	0.000668	0.001576	0.000364	0.000208	0.001730	0.003176	0.001224
component5	-0.000855	-0.000566	0.016449	0.001152	0.000685	0.000134	0.000011	-0.001005	-0.002588	-0.000288

Figure 6. Word-component matrix

	deflabel	component1	component2	component3	component4	component5	component6	component7	component8	component9
0	chikungunya	0.582706	0.009915	-0.025547	-0.065935	0.002809	-0.046121	0.118811	0.016777	-0.020070
1	human granulocytic anaplasmosis	0.555065	-0.081058	-0.072906	-0.000042	-0.009059	0.006425	0.024186	0.008590	-0.019471
2	human monocytic ehrlichiosis	0.405669	-0.027663	-0.066870	-0.018683	0.002116	-0.013837	0.062343	-0.008422	-0.020878
3	African tick-bite fever	0.407884	-0.118609	-0.043003	0.028166	-0.013565	0.008625	0.027334	0.041828	-0.023204
4	Astrakhan spotted fever	0.428987	-0.131661	-0.048310	0.036538	-0.016526	0.020339	0.011358	0.038068	-0.022694

Figure 7. Component-disease matrix

2.4. Retrieval algorithm

The human disease search engine performs a full-text search based on the search query to find out the diseases containing disease factors in the search query. Then the search engine sums the score of words in the search query on each row of the word-component matrix to get the row with the highest total score, and that row is also the row of the component that best matches the search query. Algorithm 1 presents the detailed working of the proposed model.

Algorithm 1 Retrieval
Input: word-component dataframe, WCdf
Output: the component number best match with search query
1: **For each** row **in** WCdf **do**
2: S = sum the scores of words of search query on row
3: Sumlist.add(row.index, S) //row index is the component number
4: **Return** sumlist.getmax().index // return the index of the element with max value in Sumlist

Finally, the search engine analyzes the component-disease matrix and browses the column of the most suitable component for the search query found in the previous step. To achieve the same goal, Algorithm 2 is designed, which transfers the raw data into a required input format. The result diseases are ranked based on the scores of the result disease rows.

Algorithm 2 Structuring data
Input: component-disease dataframe, CDdf
the component number best match with search query, compnumber
the result disease id list from the full text search, diseaseIDlist
Output: the ranked result disease id list
1: **For each** row **in** CDdf[[‘diseaseID’,’component’+‘compnumber’]] **do**
2: **If** row.diseaseID in diseaseIDlist **then**
3: rankdiseaseIDlist.add(row.diseaseID.value, row.’component’+‘compnumber’.value)
4: **Return** rankdiseaseIDlist.sort()// return the rankdiseaseIDlist sorted by the values of the elements

3. RESULTS AND DISCUSSION

In order to investigate the effectiveness of this proposed ranking method, we analyzed 2 tests on the human disease search engine. In the first test, we randomly selected “fever” and “paralysis” symptoms to search on the search engine for diseases matching these symptoms. Table 1 shows returns the evaluation results of models.

Table 1. The search engine for diseases matching these symptoms

No	Disease label	Disease definition	Component1
1	Powassan encephalitis	A viral infectious disease that results_in inflammation located_in brain, has_material_basis_in Powassan virus, which is transmitted_by Ixodes and transmitted_by dermacentor species of ticks. The infection has_symptom headache, has_symptom fever, has_symptom vomiting, has_symptom stiff neck, has_symptom sleepiness, has_symptom breathing distress, has_symptom tremors, has_symptom confusion, has_symptom seizures, has_symptom paralysis, and has_symptom coma	0.778757
2	Rabies	A viral infectious disease that results_in inflammation located_in brain or located_in spinal cord, has_material_basis_in rabies virus, which is transmitted_by bite of an infected animal, or transmitted_by contact of mucous membranes with saliva of an infected animal. The infection has_symptom fever, has_symptom headache, has_symptom prickling or itching sensation at the site of bite, has_symptom anxiety, has_symptom confusion, has_symptom agitation, has_symptom delirium, has_symptom difficulty swallowing, has_symptom hydrophobia, and has_symptom paralysis	0.681131
3	La crosse encephalitis	A viral infectious disease that results_in inflammation located_in brain, has_material_basis_in la crosse virus, which is transmitted_by treehole mosquito, ochlerotatus triseriatus. The infection has_symptom seizures, has_symptom headache, has_symptom fever, has_symptom coma, and has_symptom paralysis	0.601544

Powassan encephalitis disease is ranked in the first place because this disease contains not only the symptoms in the search query, but also contains other symptoms relatively close in meaning to the symptoms in the search query. Rabies disease is ranked second because it contains its feature symptoms such as “hydrophobia”, “prickling or itching sensation at the site of bite” and “difficulty swallowing” but other symptoms of this disease also have the meaning close to the symptoms in the search query. La crosse

encephalitis disease has fewer symptoms close in meaning to the symptoms in the search query than powassan encephalitis disease and rabies disease, so it is ranked in the last place.

In the second test, we randomly selected “fever”, “sore throat” and “skin” symptoms to search on the search engine for diseases suitable to these symptoms. Table 2 shows the evaluation results. “Hand, foot, and mouth” disease is ranked first because this disease not only contains the symptoms in the search query but also contains many other symptoms that are close in meaning to the symptoms in the search query. Chickenpox disease contains fewer symptoms close to the symptoms in the search query, so it is ranked second. The analysis of the result tables shows that this proposed ranking method gives reasonable and effective results. This method not only relies on the symptoms in the result diseases matching with the symptoms in the search query, but also exploits the meaning of other symptoms in the result diseases compared to the meaning of the symptoms in the search query so that the ranking of the results is reasonable.

Table 2. The search engine for diseases suitable with these symptoms

No	Disease label	Disease definition	Component7
1	Hand, foot and mouth disease	A viral infectious disease that results_in infection located_in skin, has_material_basis_in human coxsackievirus A16 or has_material_basis_in human enterovirus 71, which are transmitted_by contaminated fomites, and transmitted_by contact with nose and throat secretions, saliva, blister fluid and stool of infected persons. The infection has_symptom fever, has_symptom poor appetite, has_symptom malaise, has_symptom sore throat, has_symptom painful sores in the mouth, and has_symptom skin rash on the palms of the hands and soles of the feet	0.181783
2	Chickenpox	A viral infectious disease that results_in infection located_in skin, has_material_basis_in human herpesvirus 3, which is transmitted_by direct contact with secretions from the rash or transmitted_by droplet spread of respiratory secretions. The infection has_symptom anorexia, has_symptom myalgia, has_symptom nausea, has_symptom fever, has_symptom headache, has_symptom sore throat, and has_symptom blisters	0.114284

4. CONCLUSION

This paper uses the LSA method to rank disease results of the human disease search engine. This method takes advantage of both the TF-IDF score and the implicit relationship between disease factors. This makes the ranking of disease results more reasonable and better. Further, this method can also be combined with deep learning techniques to adjust the results more accurately. It helps the self-diagnosis of human disease search engine be more effective.

REFERENCES

[1] R. Dragusin *et al.*, “FindZebra: a search engine for rare diseases,” *International Journal of Medical Informatics*, vol. 82, no. 6, pp. 528–538, 2013, doi: 10.1016/j.ijmedinf.2013.01.005.

[2] J. H. Joloudari, H. Saadatfar, A. Dehzangi, and S. Shamshirband, “Computer-aided decision-making for predicting liver disease using PSO-based optimized SVM with feature selection,” *Informatics in Medicine Unlocked*, vol. 17, pp. 1–30, 2019, doi: 10.1016/j.imu.2019.100255.

[3] G. Luo, “Intelligent output interface for intelligent medical search engine,” in *Proceedings of the National Conference on Artificial Intelligence*, 2008, vol. 2, pp. 1201–1206.

[4] D. Faroo, “Search engine optimization for medical publishing,” *Reconstructive Review*, vol. 7, no. 4, pp. 39–41, 2018, doi: 10.15438/rr.7.4.198.

[5] S. Köhler *et al.*, “Clinical diagnostics in human genetics with semantic similarity searches in ontologies,” *The American Journal of Human Genetics*, vol. 85, no. 4, pp. 457–464, 2009, doi: 10.1016/j.ajhg.2009.09.003.

[6] O. Winther, D. Svenstrup, P. P. Henningsen, R. Kristiásson, and H. L. Jørgensen, “FindZebra-the search engine for difficult medical cases,” *Orphanet Journal of Rare Diseases*, vol. 9, 2014, doi: 10.1186/1750-1172-9-S1-O5.

[7] K. Papineni, “Why inverse document frequency?,” in *Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001-NAACL ’01*, 2001, pp. 1–8, doi: 10.3115/1073336.1073340.

[8] J. N. Singh and S. K. Dwivedi, “Comparative analysis of IDF methods to determine word relevance in web document,” *International Journal of Computer Science Issues (IJCSI)*, vol. 11, no. 1, pp. 59–65, 2014.

[9] J. G. M. Decleer, “Introduction to industrial minerals,” *Applied Clay Science*, vol. 10, no. 6, pp. 491–493, 1996, doi: 10.1016/0169-1317(95)00043-7.

[10] H. S. Nguyen, M. H. Le, C. Q. L. Lam, and T. H. Duong, “Smart interactive search for Vietnamese disease by using data mining-based ontology,” *Journal of Information and Telecommunication*, vol. 1, no. 2, pp. 176–191, 2017, doi: 10.1080/24751839.2017.1323491.

[11] “Disease ontology,” *Disease ontology*, [Online]. Available: http://disease-ontology.org/. Access date:Jan 1, 2020.

[12] L. M. Schriml *et al.*, “Disease ontology: a backbone for disease semantic integration,” *Nucleic Acids Research*, vol. 40, no. 1, pp. 940–946, 2012, doi: 10.1093/nar/gkr972.

[13] T. K. Landauer and S. T. Dumais, “A solution to plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.,” *Psychological Review*, vol. 104, no. 2, pp. 211–240, 1997, doi: 10.1037/0033-295X.104.2.211.

[14] T. K. Landauer, P. W. Foltz, and D. Laham, “An introduction to latent semantic analysis,” *Discourse Processes*, vol. 25, no. 2–3, pp. 259–284, 1998, doi: 10.1080/01638539809545028.

[15] A. Sidorova, N. Evangelopoulos, J. S. Valacich, and T. Ramakrishnan, “Uncovering the intellectual core of the information systems discipline,” *MIS Quarterly*, vol. 32, no. 3, pp. 467–482, 2008, doi: 10.2307/25148852.

[16] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990, doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[17] D. Ward, J. Hahn, and K. Feist, “Autocomplete as Research Tool: A Study on Providing Search Suggestions”, *Information Technology and Libraries*, vol. 31, no. 4, pp. 6-19, Dec. 2012, https://doi.org/10.6017/ital.v31i4.1930

[18] G. Luo, “Lessons learned from building the iMED intelligent medical search engine,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 5138–5142, doi: 10.1109/IEMBS.2009.5334577.

[19] S. Bauer, S. Kohler, M. H. Schulz, and P. N. Robinson, “Bayesian ontology querying for accurate and noise-tolerant semantic searches,” *Bioinformatics*, vol. 28, no. 19, pp. 2502–2508, 2012, doi: 10.1093/bioinformatics/bts471.

[20] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988, doi: 10.1016/0306-4573(88)90021-0.

[21] S. Jabri, A. Dahbi, T. Gadi, and A. Bassir, “Ranking of text documents using TF-IDF weighting and association rules mining,” in *2018 4th International Conference on Optimization and Applications (ICOA)*, 2018, pp. 1–6, doi: 10.1109/ICOA.2018.8370597.

[22] M. D. Gordon and S. Dumais, “Using latent semantic indexing for literature based discovery,” *Journal of the American Society for Information Science*, vol. 49, no. 8, pp. 674–685, 1998, doi: 10.1002/(SICI)1097-4571(199806)49:8<674::AID-ASI2>3.0.CO;2-T.

[23] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000, doi: 10.1126/science.290.5500.2323.

[24] X. Zhang *et al.*, “A multi-view ensemble classification for clinically actionable genetic mutations,” in *The NIPS ’17 Competition: Building Intelligent Systems*, Cham: Springer, 2018, pp. 79–99, doi: 10.1007/978-3-319-94042-7_5.

[25] M. W. Berry, S. T. Dumais, and G. W. O’Brien, “Using linear algebra for intelligent information retrieval,” *SIAM Review*, vol. 37, no. 4, pp. 573–595, 1995, doi: 10.1137/1037127.

BIOGRAPHIES OF AUTHORS

Loi Chan Quan Lam is currently pursuing Ph.D degree of Computer Science at VSB—Technical University of Ostrava, Czech Republic. In 2013, he received a half full scholarship to the master's program of International University-VNU and then he graduated with a master's degree in Information Technology Management. In 2007, he received a bachelor's degree in Information Technology from the University of Natural Sciences-VNU. He also won the english training scholarship of project 322 of the Vietnamese Ministry of Education in 2008. He had many years working about testing, software development for major companies. His main interests are data mining and machine learning. He can be contacted at email: lam.chan.quan.loi.st@vsb.cz.

Tran Kim Toai received a Bachelor's degree in information technology management and a Master's degree in business administration. He is pursuing a Ph.D degree with the Department of Computer Science, VSB–Technical of Ostrava, Czech Republic. He works as a lecturer at the Ho Chi Minh University of Technology and Education. His academic research interests are machine learning, artificial intelligence, data mining, neural network, data analysis, and finance. He has been a first author or co-author of many academic publications, including papers in Scopus Indexed journals, the lecture notes in artificial intelligence, and the book series from Springer, IEEE Xplore, and SMC-sponsored international conferences. Several academic articles have appeared in the well-respected journal articles mentioned above. He has worked in major companies and universities for 18 years involving e-commerce, databases, networking, and cyber security. His practical experience allows him to deal effectively with various issues in corporate and university working environments. He can be contacted at email: toaitk@hcmute.edu.vn.

Vaclav Snasel is currently a Professor with the Department of Computer Science, VSB—Technical University of Ostrava, Czech Republic. He works as a researcher and a university teacher. He is also the Dean of the Faculty of Electrical Engineering and the Department of Computer Science. He is the head of the research programme IT4 knowledge management, European center of excellence IT4 innovations. His research and development experience includes over 30 years in the industry and academia. He works in a multi-disciplinary environment involving artificial intelligence, social networks, conceptual lattice, information retrieval, semantic web, knowledge management, data compression, machine intelligence, neural networks, web intelligence, nature and bio-inspired computing, data mining, and applied to various real-world problems. He can be contacted at email: vaclav.snasel@vsb.cz.